

function calls with XML over a TCP/IP connection (though, an alternative protocol, such as HTTP can be used instead of the ORB calls). Moreover, a format other than XML can be used to transmit data and requests between the manager and agents. An abbreviated example of XML contained in an agent's response to a request from the SAN manager is provided below:

```

5      <?xml version="1.0"?>

      <DOCTYPE LegacyXml SYSTEM "legacy.dtd">

      <LegacyXml>

      <SystemXml>

      <UniqueIdXml>SystemXml:SystemXml:saigon.sanjose.ibm.com</UniqueIdXml>

10     <ParameterXml>

      <NameXml>Hostname</NameXml>

      <ValueXml>saigon.sanjose.ibm.com</ValueXml>

      </ParameterXml>

      <ParameterXml>

15     <NameXml>IP Address</NameXml>

      <ValueXml>9.113.212.78</ValueXml>

      </ParameterXml>

```

FIGURE 6 schematically illustrates the architecture of an exemplary manager 20. The manager 20 includes a SAN Manager Service module 38 that (a) effects decisions (e.g., host-to-storage device assignment) on behalf of the SAN in view of policy established by the operator/administrator; (b) correlates the aforementioned inband and outband data into a single composite view (e.g., component makeup and topology) of the SAN, and (c) serves as a primary interface to the administrator and to other applications.

LUN/SAN Topology Discovery

SAN Manager Service 38 assigns tasks to the illustrated engines, such as, discover engine or engines 40, and reassigns the assigned tasks, if needed, based on changes, e.g., in the interconnect fabric components, services load and operator/administrator requests. Further, the SAN Manager Service 38 performs the aforementioned correlation function. For example, as discussed in more detail below, each discover engine 40 can provide a portion of information regarding the topology of the SAN based on its scope. Some of this information may overlap information provided by other discover engines or may complement it. For example, a host may contain Fiber channel (FC) host bus adapters (HBA) and SSA HBA. Consequently, both the FC discover engine and the SSA discover engine can detect and report information regarding this host. The Manager Service 38 collates such fragmentary pieces of information received from the various discover engines to obtain a composite image of the topology of the SAN.

In addition to creating a composite image of the SAN, the SAN Manager Service 38 provides a high level interface with other applications for accessing this composite image. Thus, the SAN

Manager 'owns' the objects in the composite image and provides references that other applications can utilize to access these objects, such as a reference to the fabric level objects which contain the component objects.

- 5 With continuing reference to FIGURE 6, the SAN manager 20 includes one or more fiber channel (FC) discover engines, such as the discover engine 40 responsible for gathering topology and attribute information for the SAN components. The FC discover engine is subdivided into the following functional areas: (1) *Control*: which coordinates the activity of the other areas; (2) *Correlations*: which pulls together the information from various subprocesses and creates a composite image within the scope of a single discover engine, and (c) *Attributes*: which processes the information from various attribute scanners, as described in more detail below (in addition to processing attribute information from upper level protocol commands utilized by the scanners, the attribute processor also identifies some topology information based on inferences from the devices available to the host systems); (4) *Topology*: which processes the information from the Topology scanners (inband and outband).

The discover engine 40 receives and processes information gathered by one or more scanners, such as scanner 42, which are executables that interact with the hosts by performing system calls and IOCTL calls to gather information. Since each scanner needs to directly interact with the operating system of the host on which it resides, each scanner is custom to the operating system of its host, and hence may not be portable. To restrict this non-portability, each scanner runs within an environment set up by a Scanner Subagent, such as exemplary subagent 44, and returns information to the subagent, which in turn forwards the information to other services.